

1 In the Specification:

2 Please amend the first full paragraph on page 1 (lines 2-3) of the
3 specification as follows:

4 This patent application is related to co-pending U.S. Patent
5 Application _____, (Attorney's Docket Numbers: MS1-711US).
6 Application No. 09/756,052, filed January 5, 2001, entitled "Methods and
7 Arrangements for Providing Improved Software Version Control in
8 Managed Devices."

9
10 Please amend the last paragraph on page 1 (lines 22-24) continuing onto
11 page 2 (lines 1-2) of the specification as follows:

12
13 It would be beneficial to be able to upgrade the down loader code
14 within a managed device. Hence, there is a need for improved methods and
15 arrangements for updating the software in such managed devices.
16 Preferably, the methods and arrangements will [[be]] allow for a failsafe
17 upgrade to all of the code in the managed device, including any code used
18 to provide such upgrades.

19
20 Please amend the third full paragraph on page 4 (lines 12-17) of the
21 specification as follows:

22
23 Therefore, the goal of an upgrade process design is to insure ensure
24 that there is always a way to download new upgrades even if the last
25

upgrade failed. The conventional approach to this problem is to reserve some of the data storage space, and place a permanent down loading software (hereinafter, referred to generally as a "down loader") in it. Here, the down loader is a separate piece of software that is not used during the normal operation of the managed device.

Please amend the last paragraph on page 4 (starting on line 18 and continuing through line 14 on page 5) of the specification as follows:

There are several drawbacks with this technique, however. First, [[The]] the down loader code can be quite significant in size. In the MSN Companion, for example, the down loader includes basic kernel code, file system drivers, ~~TCL/IP~~ TCP/IP stack code, communication device drivers, WinINET code, security, and display drivers. Adding more storage to such devices can be prohibitively expensive, since it would likely require additional persistent memory (e.g., FLASH memory). Secondly, the down loader itself in such a configuration cannot be upgraded. This could present a problem in the long run since, for example, the communication protocols, which the down loader uses, may need to change over time to keep up with the evolving communications technology and/or infrastructure. Thus, keeping the down loader fixed forever would be troublesome. For managed devices connected to the Internet, it may also be desirable to change Internet service providers (ISPs) from time to time. A fixed down loader, however, could make changing ISPs difficult, if not impossible. Thirdly, a fixed down loader ~~simply~~ simply cannot provide newer and potentially

richer user interactivity features that enhance the upgrade process. To keep the down loader's size to a minimum, designers usually have to reduce the number and variety of such user features. For example, for certain managed devices it would be nice to show the user information about any new enhancements during the often long upgrade download. The code required to provide this and other capabilities tends to be too large in size to provide in a typical fixed down loader.

BEST AVAILABLE COPY